Published before January 2001
Terms used **shared class** and **handle**

Found **15** of **112,783**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 15 of 15

Relevance scale

1 Minimum area retiming with equivalent initial states
Naresh Maheshwari, Sachin S. Sapatnekar
November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(105.24 KB) Publisher Site Additional Information: full citation, abstract, references, citings, index terms

Traditional minimum area retiming algorithms attempt to achieve their prescribed objective with no regard to maintaining the initial state of the system. This issue is important for circuits such as controllers, and our work addresses this problem. The procedure described generates bounds on the retiming variables that guarantee an equivalent initial state after retiming. A number of possible sets of bounds can be derived, and each set is used to solve a minimum area retiming problem that is set ...

Keywords: Retiming, Sequential Circuits, VLSI, Design Automation, Timing Optimization, Area Optimization

2 Phoenix architecture
John Roder
June 1978 **ACM SIGDA Newsletter**, Volume 8 Issue 2
Full text available: pdf(472.15 KB) Additional Information: full citation, abstract

Multiprogramming and multitasking programming concepts are familiar to most of us. In a multiprogrammed arrangement several "user programs" can share the use of a single computer in a time sharing mode. An executive program provides the time slices to the "user programs" by activating and deactivating them according to some schedule. When multitasking is added to this arrangement, a "user program" may have several incarnations and the executive provides time slices to the incarnations of the use ...

3 Bifurcated routing in computer networks
Wai Sum Lai
July 1985 **ACM SIGCOMM Computer Communication Review**, Volume 15 Issue 3
Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references

This paper presents a characterization and a survey of multiple path routing in computer networks. It also develops a routing protocol that achieves load sharing and combines the

strengths of both virtual circuit and datagram networks.

4 Distributed form management
Heikki Hämmäinen, Eero Eloranta, Jari Alasuvanto
January 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 1

Full text available: pdf(2.24 MB) Additional Information: full citation, abstract, references, citings, index terms

An open architecture for distributed form management is described. The model employs object-orientation in describing organizational units as well as individual users as entities with uniform external interfaces. Each entity is represented by an autonomous user agent which operates on local and migrating forms. The form concept encapsulates data, layout, and rules into a unified object which is the basic unit of presentation, processing, storage, and commun ...

5 The DIAMOND security policy for object-oriented databases
Linda M. Null, Johnny Wong
April 1992 **Proceedings of the 1992 ACM annual conference on Communications**

Full text available: pdf(792.69 KB) Additional Information: full citation, abstract, references, citings, index terms

A formal data model and integrated multilevel security policy for object-oriented database systems are presented in this paper. The security policy addresses mandatory as well as discretionary security controls. Classes derive security classification constraints from their instances and logical instances.

6 The grok project data structures and process communication
Peter Jensen
January 1973 **ACM SIGPLAN Notices , Proceeding of ACM SIGPLAN - SIGOPS interface meeting on Programming languages - operating systems**, Volume 8 Issue 9

Full text available: pdf(386.93 KB) Additional Information: full citation, abstract, references, citings, index terms

An overview of the structured data types proposed for the grok language is given. The proposal represents a uniform approach to arrays, records and procedures by regarding them as specialized occurences of the same underlying structure. Structures are partitioned into three classes according to the number of references pointing to the structure at the same time, and we show how this may be utilised in synchronization of parallel processes. The well known producer-consumer example illustrate ...

7 Queueing Network Modeling of Computer Communication Networks
J. W. Wong
September 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 3

Full text available: pdf(650.65 KB) Additional Information: full citation, references, citings, index terms

8 The design, implementation and evaluation of SMART: a scheduler for multimedia applications
Jason Nieh, Monica S. Lam
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5

Full text available: pdf(2.48 MB) Additional Information: full citation, references, citings, index terms

9 Class management for software communities

Simon Gibbs, Eduardo Casais, Oscar Nierstrasz, X. Pintado, Dennis Tsichritzis
September 1990 **Communications of the ACM**, Volume 33 Issue 9

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Object-oriented programming may engender an approach to software development characterized by the large-scale reuse of object classes. Large-scale reuse is the use of a class not just by its original developers, but by other developers who may be from other organizations, and may use the classes over a long period of time. Our hypothesis is that the successful dissemination and reuse of classes requires a well-organized community of developers who are ready to share ideas, methods, tools an ...

10 Simulation of dispatching algorithms in a multiprogramming environment

N. S. Losapio, William G. Bulgren
August 1972 **Proceedings of the ACM annual conference - Volume 2**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, index terms

CPU scheduling or dispatching in computer systems is surveyed. In order to study and evaluate a dispatching algorithm, a simulation model of a dispatching algorithm for multiprogramming operating system has been devised and programmed. Essential elements which have been included are the job environment, the computer system environment, and the user environment. The model is patterned basically after GECOS III, on the H-600 line computer. Some results of the investigation are presented, such ...

Keywords: CPU scheduling, Computer system analysis, Multiprogramming, Simulation

11 Design components: toward software composition at the design level

Rudolf K. Keller, Reinhard Schauer
April 1998 **Proceedings of the 20th international conference on Software engineering**

Full text available: pdf(1.33 MB) Publisher Site Additional Information: full citation, references, citings, index terms

12 The new crop of Java virtual machines (panel)

Lars Bak, John Duimovich, Jesse Fang, Scott Meyer, David Ungar
October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10

Full text available: pdf(399.37 KB) Additional Information: full citation, citings, index terms

13 What have we learnt from using real parallel machines to solve real problems?

G. C. Fox
January 1989 **Proceedings of the third conference on Hypercube concurrent computers and applications - Volume 2**

Full text available: pdf(4.08 MB) Additional Information: full citation, abstract, references, citings, index terms

We briefly review some key scientific and parallel processing issues in a selection of some 84 existing applications of parallel machines. We include the MIMD hypercube transputer array, BBN Butterfly, and the SIMD ICL DAP, Goodyear MPP and Connection Machine from Thinking Machines. We use a space-time analogy to classify problems and show how a

division into synchronous, loosely synchronous and asynchronous problems is helpful. This classifies problems into those suitable for SIMD or MIMD ...

14 An efficient implementation of SELF a dynamically-typed object-oriented language based on prototypes

C. Chambers, D. Ungar, E. Lee

September 1989 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 24 Issue 10

Full text available: pdf(2.41 MB) Additional Information: full citation, abstract, references, citings, index terms

We have developed and implemented techniques that double the performance of dynamically-typed object-oriented languages. Our SELF implementation runs twice as fast as the fastest Smalltalk implementation, despite SELF's lack of classes and explicit variables. To compensate for the absence of classes, our system uses implementation-level maps to transparently group objects cloned from the same prototype, providing data type information and eliminating the apparent ...

15 Navigation issues in hypertext: documenting complex hierarchies with HTML frames

Michael Priestley

October 1997 **Proceedings of the 15th annual international conference on Computer documentation**

Full text available: pdf(1.35 MB) Additional Information: full citation, references, citings, index terms

Results 1 - 15 of 15

Published before January 2001
Terms used **shared class**

Found **37** of **112,783**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 37 Result page: **1** 2 next

Relevance scale

1 Minimum area retiming with equivalent initial states
Naresh Maheshwari, Sachin S. Sapatnekar
November 1997 **Proceedings of the 1997 IEEE/ACM international conference on Computer-aided design**
Full text available: pdf(105.24 KB) Publisher Site
Additional Information: full citation, abstract, references, citings, index terms

Traditional minimum area retiming algorithms attempt to achieve their prescribed objective with no regard to maintaining the initial state of the system. This issue is important for circuits such as controllers, and our work addresses this problem. The procedure described generates bounds on the retiming variables that guarantee an equivalent initial state after retiming. A number of possible sets of bounds can be derived, and each set is used to solve a minimum area retiming problem that is set ...

Keywords: Retiming, Sequential Circuits, VLSI, Design Automation, Timing Optimization, Area Optimization

2 Java's insecure parallelism
Per Brinch Hansen
April 1999 **ACM SIGPLAN Notices**, Volume 34 Issue 4
Full text available: pdf(432.76 KB) Additional Information: full citation, abstract, citings, index terms

The author examines the synchronization features of Java and finds that they are insecure variants of his earliest ideas in parallel programming published in 1972-73. The claim that Java supports monitors is shown to be false. The author concludes that Java ignores the last twenty-five years of research in parallel programming languages.

Keywords: Java, monitors, parallel programming, programming languages, security

3 The DIAMOND security policy for object-oriented databases
Linda M. Null, Johnny Wong
April 1992 **Proceedings of the 1992 ACM annual conference on Communications**
Full text available: pdf(792.69 KB) Additional Information: full citation, abstract, references, citings, index terms

A formal data model and integrated multilevel security policy for object-oriented database systems are presented in this paper. The security policy addresses mandatory as well as discretionary security controls. Classes derive security classification constraints from their instances and logical instances.

4 Distributed form management

Heikki Hämmäinen, Eero Eloranta, Jari Alasuvanto

January 1990 **ACM Transactions on Information Systems (TOIS)**, Volume 8 Issue 1

Full text available: pdf(2.24 MB) Additional Information: full citation, abstract, references, citings, index terms

An open architecture for distributed form management is described. The model employs object-orientation in describing organizational units as well as individual users as entities with uniform external interfaces. Each entity is represented by an autonomous user agent which operates on local and migrating forms. The form concept encapsulates data, layout, and rules into a unified object which is the basic unit of presentation, processing, storage, and commun ...

5 An evaluation of the real-time performances of SVR4.0 and SVR4.2

Sherali Zeadally

January 1997 **ACM SIGOPS Operating Systems Review**, Volume 31 Issue 1

Full text available: pdf(583.94 KB) Additional Information: full citation, abstract, index terms

UNIX is one of the most widely used operating systems on current workstations. However, UNIX was originally designed as a multitasking and time-sharing system with little concern for supporting real-time applications. Recent versions of UNIX have incorporated real-time features and the designers of these systems claim to provide better response times than the standard UNIX kernel. In order to assess the benefits of these new features and verify these claims, this paper compares the real-time per ...

6 DRAGOON: a tool for the Ada programmer

Stephen J. Goldsack, Colin Atkinson

December 1991 **Proceedings of the conference on TRI-Ada '91: today's accomplishments; tomorrow's expectations**

Full text available: pdf(1.17 MB) Additional Information: full citation, references, citings, index terms

7 Bifurcated routing in computer networks

Wai Sum Lai

July 1985 **ACM SIGCOMM Computer Communication Review**, Volume 15 Issue 3

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, references

This paper presents a characterization and a survey of multiple path routing in computer networks. It also develops a routing protocol that achieves load sharing and combines the strengths of both virtual circuit and datagram networks.

8 Queueing Network Modeling of Computer Communication Networks

J. W. Wong

September 1978 **ACM Computing Surveys (CSUR)**, Volume 10 Issue 3

Full text available: pdf(650.65 KB) Additional Information: full citation, references, citings, index terms

9 Type-based race detection for Java

Cormac Flanagan, Stephen N. Freund
May 2000 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2000 conference on Programming language design and implementation**, Volume 35 Issue 5

Full text available: pdf(237.37 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents a static race detection analysis for multithreaded Java programs. Our analysis is based on a formal type system that is capable of capturing many common synchronization patterns. These patterns include classes with internal synchronization, classes thatrequire client-side synchronization, and thread-local classes. Experience checking over 40,000 lines of Java code with the type system demonstrates that it is an effective approach for eliminating races conditions. On lar ...

10 Connecting viewpoints by shared phenomena
Michael Jackson
October 1996 **Joint proceedings of the second international software architecture workshop (ISAW-2) and international workshop on multiple perspectives in software development (Viewpoints '96) on SIGSOFT '96 workshops**

Full text available: pdf(583.21 KB) Additional Information: full citation, references, index terms

11 Link-sharing and resource management models for packet networks
Sally Floyd, Van Jacobson
August 1995 **IEEE/ACM Transactions on Networking (TON)**, Volume 3 Issue 4

Full text available: pdf(2.51 MB) Additional Information: full citation, references, citings, index terms

12 Guarded methods vs. inheritance anomaly: inheritance anomaly solved by nested guarded method calls
Szabolcs Ferenczi
February 1995 **ACM SIGPLAN Notices**, Volume 30 Issue 2

Full text available: pdf(913.98 KB) Additional Information: full citation, abstract, citings, index terms

The term Inheritance Anomaly has been introduced into Object-Oriented Concurrent Programming (OOCP). It has also been shown how different kinds of synchronization mechanisms fall into one of the three anomalies. Guarded Methods are shown to be an inadequate tool to solve either History-only Sensitive Anomaly or Modification of Acceptable States. This paper argues that any of the anomalies can be solved by Guarded Methods when a suitable semantics is found for it. The main contribution of the pap ...

13 Release-to-release binary compatibility in SOM
Ira R. Forman, Michael H. Conner, Scott H. Danforth, Larry K. Raper
October 1995 **ACM SIGPLAN Notices , Proceedings of the tenth annual conference on Object-oriented programming systems, languages, and applications**, Volume 30 Issue 10

Full text available: pdf(1.66 MB) Additional Information: full citation, abstract, references, citings, index terms

SOM (IBM's System Object Model) removes a major impediment to reuse in Object-Oriented Programming by facilitating the programming of release-to-release binary compatible class libraries. This is accomplished by supporting a large number of compatibility preserving transformations. Taken together these transformations compose a discipline for programming evolving class libraries.

14 Class management for software communities

Simon Gibbs, Eduardo Casais, Oscar Nierstrasz, X. Pintado, Dennis Tsichritzis
September 1990 **Communications of the ACM**, Volume 33 Issue 9

Full text available: pdf(2.01 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Object-oriented programming may engender an approach to software development characterized by the large-scale reuse of object classes. Large-scale reuse is the use of a class not just by its original developers, but by other developers who may be from other organizations, and may use the classes over a long period of time. Our hypothesis is that the successful dissemination and reuse of classes requires a well-organized community of developers who are ready to share ideas, methods, tools an ...

15 The grok project data structures and process communication
Peter Jensen
January 1973 **ACM SIGPLAN Notices , Proceeding of ACM SIGPLAN - SIGOPS interface meeting on Programming languages - operating systems**, Volume 8 Issue 9

Full text available: pdf(386.93 KB) Additional Information: full citation, abstract, references, citings, index terms

An overview of the structured data types proposed for the grok language is given. The proposal represents a uniform approach to arrays, records and procedures by regarding them as specialized occurences of the same underlying structure. Structures are partitioned into three classes according to the number of references pointing to the structure at the same time, and we show how this may be utilised in synchronization of parallel processes. The well known producer-consumer example illustrate ...

16 Simulation of dispatching algorithms in a multiprogramming environment
N. S. Losapio, William G. Bulgren
August 1972 **Proceedings of the ACM annual conference - Volume 2**

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, references, index terms

CPU scheduling or dispatching in computer systems is surveyed. In order to study and evaluate a dispatching algorithm, a simulation model of a dispatching algorithm for multiprogramming operating system has been devised and programmed. Essential elements which have been included are the job environment, the computer system environment, and the user environment. The model is patterned basically after GECOS III, on the H-600 line computer. Some results of the investigation are presented, such ...

Keywords: CPU scheduling, Computer system analysis, Multiprogramming, Simulation

17 Sustaining software interoperability via shared, evolving object repositories: system optimization and evaluation
Viviane M. Crestana, Amy J. Lee, Elke A. Rundensteiner
November 1996 **Proceedings of the 1996 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(215.43 KB) Additional Information: full citation, abstract, references, index terms

Powerful interoperability-enabling solutions for software application integration must allow applications to evolve and data requirements to change, while minimizing such changes on other integrated applications. Thus, we have developed the transparent schema evolution (TSE) system that accomplishes evolution by generating a new object-oriented view schema to capture the changes desired by the user, while preserving existing view schemas for old applications. This generation of a potentially lar ...

18 An efficient implementation of SELF a dynamically-typed object-oriented language

based on prototypes
C. Chambers, D. Ungar, E. Lee
September 1989 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 24 Issue 10

Full text available: pdf(2.41 MB) Additional Information: full citation, abstract, references, citings, index terms

We have developed and implemented techniques that double the performance of dynamically-typed object-oriented languages. Our SELF implementation runs twice as fast as the fastest Smalltalk implementation, despite SELF's lack of classes and explicit variables. To compensate for the absence of classes, our system uses implementation-level maps to transparently group objects cloned from the same prototype, providing data type information and eliminating the apparent ...

19 What have we learnt from using real parallel machines to solve real problems?
G. C. Fox
January 1989 **Proceedings of the third conference on Hypercube concurrent computers and applications - Volume 2**

Full text available: pdf(4.08 MB) Additional Information: full citation, abstract, references, citings, index terms

We briefly review some key scientific and parallel processing issues in a selection of some 84 existing applications of parallel machines. We include the MIMD hypercube transputer array, BBN Butterfly, and the SIMD ICL DAP, Goodyear MPP and Connection Machine from Thinking Machines. We use a space-time analogy to classify problems and show how a division into synchronous, loosely synchronous and asynchronous problems is helpful. This classifies problems into those suitable for SIMD or MIMD ...

20 Quicksilver: a quasi-static compiler for Java
Mauricio Serrano, Rajesh Bordawekar, Sam Midkiff, Manish Gupta
October 2000 **ACM SIGPLAN Notices , Proceedings of the 15th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 35 Issue 10

Full text available: pdf(748.42 KB) Additional Information: full citation, abstract, references, citings, index terms

This paper presents the design and implementation of the Quicksilver[1] quasi-static compiler for Java. Quasi-static compilation is a new approach that combines the benefits of static and dynamic compilation, while maintaining compliance with the Java standard, including support of its dynamic features. A quasi-static compiler relies on the generation and reuse of persistent code images to reduce the overhead of compilation during program execution, and to provide identical, testable an ...

Results 1 - 20 of 37 Result page: **1** 2 next

Published before January 2001
Terms used **shared class**

Found **37** of **112,783**

Sort results by: relevance
Display results: expanded form

Save results to a Binder
Search Tips
Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 21 - 37 of 37 Result page: previous 1 **2**

Relevance scale

21 Introduction to computer science: an interactive approach using ISETL
Nancy Baxter, David Hastings, Jane Hill, Peter Martin, Robert Paul
February 1990 **ACM SIGCSE Bulletin , Proceedings of the twenty-first SIGCSE technical symposium on Computer science education**, Volume 22 Issue 1
Full text available: pdf(321.62 KB) Additional Information: full citation, references, citings, index terms

22 Design components: toward software composition at the design level
Rudolf K. Keller, Reinhard Schauer
April 1998 **Proceedings of the 20th international conference on Software engineering**
Full text available: pdf(1.33 MB) Publisher Site Additional Information: full citation, references, citings, index terms

23 The case for services over cascaded networks
Anthony D. Joseph, B. R. Badrinath, Randy H. Katz
October 1998 **Proceedings of the 1st ACM international workshop on Wireless mobile multimedia**
Full text available: pdf(1.08 MB) Additional Information: full citation, references, citings, index terms

24 Scheduling with implicit information in distributed systems
Andrea C. Arpaci-Dusseau, David E. Culler, Alan M. Mainwaring
June 1998 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1998 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems**, Volume 26 Issue 1
Full text available: pdf(1.63 MB) Additional Information: full citation, abstract, references, citings, index terms

Implicit coscheduling is a distributed algorithm for time-sharing communicating processes in a cluster of workstations. By observing and reacting to implicit information, local schedulers in the system make independent decisions that dynamically coordinate the scheduling of communicating processes. The principal mechanism involved is *two-phase spin-blocking*: a process waiting for a message response spins for some amount of time, and then

relinquishes the processor if the response d ...

25 The new crop of Java virtual machines (panel)
Lars Bak, John Duimovich, Jesse Fang, Scott Meyer, David Ungar
October 1998 **ACM SIGPLAN Notices , Proceedings of the 13th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 33 Issue 10
Full text available: pdf(399.37 KB) Additional Information: full citation, citings, index terms

26 The design, implementation and evaluation of SMART: a scheduler for multimedia applications
Jason Nieh, Monica S. Lam
October 1997 **ACM SIGOPS Operating Systems Review , Proceedings of the sixteenth ACM symposium on Operating systems principles**, Volume 31 Issue 5
Full text available: pdf(2.48 MB) Additional Information: full citation, references, citings, index terms

27 Targeting GNAT to the Java virtual machine
Cyrille Comar, Gary Dismukes, Franco Gasperoni
November 1997 **Proceedings of the conference on TRI-Ada '97**
Full text available: pdf(1.72 MB) Additional Information: full citation, references, citings, index terms

28 Split objects: a disciplined use of delegation within objects
Daniel Bardou, Christophe Dony
October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 31 Issue 10
Full text available: pdf(1.96 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper's primary aim is to improve the understanding of the delegation mechanism as defined in [18]. We propose a new characterization of delegation based on the notions of name sharing, property sharing and value sharing. It allows us (1) to clearly differentiate delegation from class-inheritance in particular and more generally from other inheritance mechanisms and (2) to explain how a founded use of delegation relies on a correct semantics of variable property sharing between objects conn ...

29 Code reuse in an optimizing compiler
Ali-Reza Adl-Tabatabai, Thomas Gross, Guei-Yuan Lueh
October 1996 **ACM SIGPLAN Notices , Proceedings of the 11th ACM SIGPLAN conference on Object-oriented programming, systems, languages, and applications**, Volume 31 Issue 10
Full text available: pdf(1.97 MB) Additional Information: full citation, abstract, references, citings, index terms

This paper describes how the cmcc compiler reuses code---both internally (reuse between different modules) and externally (reuse between versions for different target machines). The key to reuse are the application frameworks developed for global data-flow analysis, code generation, instruction scheduling, and register allocation.The code produced by cmcc is as good as the code produced by the native compilers for the MIPS and SPARC, although significantly less resources have been spent on cmcc ...

30 Navigation issues in hypertext: documenting complex hierarchies with HTML frames
Michael Priestley
October 1997 **Proceedings of the 15th annual international conference on Computer documentation**
Full text available: pdf(1.35 MB) Additional Information: full citation, references, citings, index terms

31 Integrating constraints in complex objects
C. Oussalah, V. Puig
November 1996 **Proceedings of the fifth international conference on Information and knowledge management**
Full text available: pdf(932.95 KB) Additional Information: full citation, references, index terms

32 Goal-oriented buffer management revisited
Kurt P. Brown, Michael J. Carey, Miron Livny
June 1996 **ACM SIGMOD Record , Proceedings of the 1996 ACM SIGMOD international conference on Management of data**, Volume 25 Issue 2
Full text available: pdf(1.56 MB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we revisit the problem of achieving multi-class workload response time goals by automatically adjusting the buffer memory allocations of each workload class. We discuss the virtues and limitations of previous work with respect to a set of criteria we lay out for judging the success of any goal-oriented resource allocation algorithm. We then introduce the concept of *hit rate concavity* and develop a new goal-oriented buffer allocation algorithm, called *Class Fencing,* th ...

33 Batch class process scheduler for Unix SVR4
Jan Braams
May 1995 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems**, Volume 23 Issue 1
Full text available: pdf(241.01 KB) Additional Information: full citation, index terms

34 The DIAMOND security policy for object-oriented databases
Linda M. Null, Johnny Wong
March 1992 **Proceedings of the 1992 ACM/SIGAPP symposium on Applied computing: technological challenges of the 1990's**
Full text available: pdf(780.49 KB) Additional Information: full citation, references, index terms

35 Concurrency annotations for reusable software
Klaus-Peter Löhr
September 1993 **Communications of the ACM**, Volume 36 Issue 9
Full text available: pdf(3.56 MB) Additional Information: full citation, references, citings, index terms

Keywords: concurrency annotations, concurrent object-oriented programming

36 Comparison of hardware and software cache coherence schemes
Sarita V. Adve, Vikram S. Adve, Mark D. Hill, Mary K. Vernon
April 1991 **ACM SIGARCH Computer Architecture News , Proceedings of the 18th annual international symposium on Computer architecture**, Volume 19 Issue 3
Full text available: pdf(1.22 MB) Additional Information: full citation, references, citings, index terms

37 Phoenix architecture
John Roder
June 1978 **ACM SIGDA Newsletter**, Volume 8 Issue 2
Full text available: pdf(472.15 KB) Additional Information: full citation, abstract

Multiprogramming and multitasking programming concepts are familiar to most of us. In a multiprogrammed arrangement several "user programs" can share the use of a single computer in a time sharing mode. An executive program provides the time slices to the "user programs" by activating and deactivating them according to some schedule. When multitasking is added to this arrangement, a "user program" may have several incarnations and the executive provides time slices to the incarnations of the use ...

Results 21 - 37 of 37 Result page: previous 1 **2**

CiteSeer Find: Documents Citations

Searching for **PHRASE shared class**.
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
36 documents found. **Order: number of citations.**

Systematic Realisation of Control Flow Analyses for CML - Gasser, Nielson, Nielson (1997) (Correct) (10 citations)
parallel threads that communicate via first **class shared** locations, and they present a 0-CFA like
www.daimi.au.dk/~fn/Papers/GNN97cml.ps

Concurrent Programming with Shared Objects in Networked.. - Hartley, Sunderam (1993) (Correct) (8 citations)

and by inheriting attributes of the DoPVM **class shared**, the programmer may build custom data
/inheriting attributes from {system defined **class shared** public: pvm_int x[100]constructing
phase.etl.go.jp/parallel/environments/pvm3/emory-vss/dopvm.ps.Z

The Object Binary Interface - C++ Objects for Evolvable.. - Goldstein, Sloane (1994) (Correct) (6 citations)
Binary Interface-CObjects for Evolvable **Shared Class** Libraries Theodore C. Goldstein Alan D.
Binary Interface-CObjects for Evolvable **Shared Class** Libraries Theodore C. Goldstein Sun
swt-www.informatik.uni-hamburg.de/~1friedri/sv/references/Evolvable_Shared_Class_Libraries.ps.gz

Static Conflict Analysis for Multi-Threaded Object-Oriented.. - von Praun, Gross (2003) (Correct) (2 citations)
sparse program instrumentations for dynamic **class Shared** {int i Shared (i =0 20) class
with two threads that both access an object of **class Shared**. We use this example to illustrate how OUGs
method Example:main. B. The object g10 of **class Shared** is created and initialized. Allocation and
www.lst.inf.ethz.ch/research/publications/./publications/PLDI_2003/PLDI_2003.ps.gz

Predictable Management of System Resources for Linux - Mansoor Alicherry Gopinath (2001) (Correct) (1 citation)
in turn have higher priority than the time **shared class**. Scheduling is done based on this priority. So
real time class, none of the system and time **shared class** processes are allowed to run. This model is
www.usenix.org/publications/library/proceedings/usenix01/freenix01/full_papers/alicherry/alicherry.ps

Writing an Active Application for the ASP Execution.. - Phillips, Braden.. (2000) (Correct) (1 citation)
methods are allowed. Locks cannot be placed on **shared class** code between processes. Applications can
www.isi.edu/active-signal/ARP/DOCUMENTS/PPI.ps

Designing Space in Virtual Environments for Aiding Wayfinding.. - Charitos (1997) (Correct) (1 citation)
Lynch's concept of landmark as identifying a **class shared** by any singular referent for thought, any
www.brunel.ac.uk/depts/mes/Research/Groups/vvr/vrsig97/postscript/028.ps.gz

Unknown - Egf Ch Ji (Correct)
of the fact that the type system supports first-**class, shared**, mutable references with strong updates. We
www.eecs.harvard.edu/%7Eamal/papers/linloc-techrpt.pdf

A Linear Language with Locations - Greg Morrisett Amal (Correct)
of the fact that the type system supports first-**class, shared**, mutable references with strong updates. We
www.eecs.harvard.edu/%7Eamal/papers/linloc.pdf

Negligent Class Loaders for Software Evolution - Yoshiki Sato And (2004) (Correct)
barrier makes it di#cult to include a copy of a **shared class** in every component. If two components share a
should be able to include a copy of that **shared class** since an ideal component should contain all
www.csg.is.titech.ac.jp/paper/yoshiki-ecoop-ram-se2004.pdf

Static Conflict Analysis for - Multi-Threaded Object-Oriented.. (Correct)
program instrumentations for dynamic 2 **class Shared** {int i Shared (i =0 20) class
with two threads that both access an object of **class Shared**. We use this example to illustrate how OUGs

method Example:main. B. The object g10 of **class Shared** is created and initialized. Allocation and
pag.lcs.mit.edu/reading-group/praun03conflict.ps

Type-Based Information Flow Analysis for the Pi-Calculus - Kobayashi (2003) (Correct)
primitives can be expressed in terms of **shared** memory primitives by using, for example,
or not. Since synchronization algorithms based on **shared** memory primitives inspect the values of **shared**
www.kb.cs.titech.ac.jp/~kobayasi/papers/iflow-pi.ps.gz

Facilitating - Successful Online Computing (Correct)
the natural reticence of students to post in **shared class** spaces. In computing courses an additional
that students are reticent about posting in **shared class** spaces-perhaps because they lack the
crpit.com/confpapers/CRPITV30Young.pdf

Using a Taxonomy Tool to Identify Changes in OO Software - Clarke, Malloy, Gibson (2003) (Correct)
how criteria such as types, accessibility, **shared class** data, deferred features, dynamic binding,
the attribute. Static -if the attribute is **shared class** data. Family NA -represents no class
www.cs.may.ie/~pgibson/Postscripts/CSMR2003.ps

Overview of Recent Supercomputers - van der Steen (1997) (Correct)
are described according to their architectural **class. Shared**- and distributed memory SIMD- and MIMD
www.euroben.nl/reports/overview94.ps.gz

Knowledge Sources for Word Sense Disambiguation - Agirre, Martinez (Correct)
in "The chair and the table were missing" the **shared class** in the taxonomy with table can be used to
ixa.si.ehu.es/dokument/Artikulu/01tsd.pdf

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP